**品优购电商系统开发**

**第5章**

**安全框架与商家入驻审核**

传智播客.黑马程序员

# 课程目标

目标1：完成运营商系统登录与安全控制功能

目标2：完成商家申请入驻的功能

目标3：完成商家审核的功能

目标4：完成商家系统登陆与安全控制功能

# 1.Spring Security回顾

## 1.1步骤一

依赖配置：

|  |
| --- |
| *<!-- spring-security -->* <**dependency**>  <**groupId**>org.springframework.security</**groupId**>  <**artifactId**>spring-security-web</**artifactId**>  <**version**>4.2.3.RELEASE</**version**> </**dependency**> <**dependency**>  <**groupId**>org.springframework.security</**groupId**>  <**artifactId**>spring-security-config</**artifactId**>  <**version**>4.2.3.RELEASE</**version**> </**dependency**> |

## 1.2步骤二

web.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0"**>  *<!-- 配置Spring加载文件 -->* <**context-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:applicationContext\*.xml</**param-value**>  </**context-param**>  *<!-- 配置Spring的核心监听器 -->* <**listener**> <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**>  </**listener**>   *<!-- 配置SpringSecurity委派过滤器 -->* <**filter**>  <**filter-name**>springSecurityFilterChain</**filter-name**>  <**filter-class**>**org.springframework.web.filter.DelegatingFilterProxy**</**filter-class**>  </**filter**>  <**filter-mapping**>  <**filter-name**>springSecurityFilterChain</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**> </**web-app**> |

## 1.3步骤三

applicationContext-security.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:security="http://www.springframework.org/schema/security"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/security  http://www.springframework.org/schema/security/spring-security.xsd"**>   *<!-- ##### 配置http请求URL拦截规则 ##### -->  <!-- 配置哪些请求URL不需要拦截 -->* <**security:http pattern="/login.html" security="none"**/>  <**security:http**>  *<!-- 配置拦截的请求URL，需要用什么角色访问 ROLE\_\* -->* <**security:intercept-url pattern="/\*\*" access="hasRole('USER')"**/>  *<!-- 配置表单登录 -->* <**security:form-login login-page="/login.html"  login-processing-url="/login"  username-parameter="username"  password-parameter="password"  default-target-url="/login\_success.html"  always-use-default-target="true"  authentication-failure-url="/login.html"**/>  *<!-- 关闭CSRF跨站点请求伪造 -->* <**security:csrf disabled="true"**/>  </**security:http**>   *<!-- ######### 配置身份认证管理器 ########## -->* <**security:authentication-manager**>  *<!-- 配置认证提供者 -->* <**security:authentication-provider**>  *<!-- 配置用户服务作为，认证的提供者 -->* <**security:user-service**>  *<!-- 配置一个用户: 用户名、密码、授权(角色) -->* <**security:user name="admin" password="123456" authorities="ROLE\_USER"**/>  </**security:user-service**>  </**security:authentication-provider**>  </**security:authentication-manager**> </**beans**> |

配置说明：

<security:intercept-url pattern=”/\*\*”/>：表示拦截的地址

/\* 表示的是该目录下的资源，只包括本级目录不包括下级目录

/\*\* 表示的是该目录以及该目录下所有级别子目录的资源

<security:form-login/>： 开启表单登录

-- login-page：指定登录页面。

-- login-processing-url: 登录处理的URL

-- password-parameter: 密码请求参数名称

-- username-parameter: 用户名请求参数名称

-- authentication-failure-url：身份验证失败时跳转的页面(重定向)

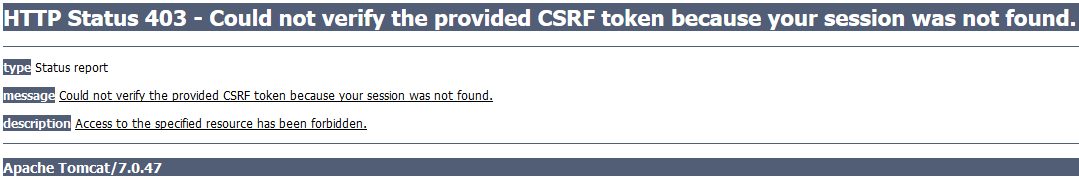
-- Authentication-failure-forward-url：身份验证失败时跳转的页面(请求转发)

-- authentication-success-forward-url：身份验证成功时跳转的页面(请求转发)

-- default-target-url：指定了成功进行身份验证和授权后默认呈现给用户的页面

-- always-use-default-target: 总是用上面默认的目标页面。

csrf disabled="true" 关闭csrf ,如果不加会出现错误



CSRF（Cross-site request forgery）跨站点请求伪造。

**CSRF攻击原理及过程如下：**

1. 用户打开浏览器，访问受信任网站A，输入用户名和密码请求登录网站A。
2. 在用户信息通过验证后，网站A产生Cookie信息并返回给浏览器，此时用户成功登录网站A，可以正常发送请求到网站A。
3. 用户未退出网站A之前，在同一浏览器中，打开一个TAB页访问网站B。
4. 网站B接收到用户请求后，返回一些攻击性代码，并发出一个请求要求访问第三方网站(站点)A。
5. 浏览器在接收到这些攻击性代码后，根据网站B的请求，在用户不知情的情况下携带Cookie信息，向网站A发出请求。网站A并不知道该请求其实是由网站B发起的，所以会根据用户的Cookie信息及用户的权限处理该请求，导致来自网站B的恶意代码被执行。

# 2.运营商系统登录与安全控制

## 2.1需求分析

完成运营商登录功能：http://manager.pinyougou.com/login.html



## 2.2登录功能的实现

### 2.2.1配置文件

修改pinyougou-manager-web的pom.xml，添加依赖

|  |
| --- |
| <**dependencies**>  *<!-- spring-security -->* <**dependency**>  <**groupId**>org.springframework.security</**groupId**>  <**artifactId**>**spring-security-web**</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>org.springframework.security</**groupId**>  <**artifactId**>**spring-security-config**</**artifactId**>  </**dependency**> </**dependencies**> |

修改pinyougou-manager-web/src/main/webapp/WEB-INF/web.xml加入配置

|  |
| --- |
| *<!-- 配置Spring加载文件 -->* <**context-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:applicationContext\*.xml</**param-value**> </**context-param**> *<!-- 配置Spring的核心监听器 -->* <**listener**> <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**> </**listener**> *<!-- 配置SpringSecurity委派过滤器 -->* <**filter**>  <**filter-name**>springSecurityFilterChain</**filter-name**> <**filter-class**>org.springframework.web.filter.DelegatingFilterProxy</**filter-class**> </**filter**> <**filter-mapping**>  <**filter-name**>springSecurityFilterChain</**filter-name**>  <**url-pattern**>/\*</**url-pattern**> </**filter-mapping**> |

在pinyougou-manager-web/src/main/resources/目录下创建

applicationContext-security.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:security="http://www.springframework.org/schema/security"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/security  http://www.springframework.org/schema/security/spring-security.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd"**>   *<!--########## 配置请求URL拦截规则 ##########-->  <!-- 配置哪些请求URL不需要拦截 -->* <**security:http pattern="/login.html" security="none"**/>  <**security:http pattern="/css/\*\*" security="none"**/>  <**security:http pattern="/img/\*\*" security="none"**/>  <**security:http pattern="/js/\*\*" security="none"**/>  <**security:http pattern="/plugins/\*\*" security="none"**/>   *<!-- 配置哪些请求URL需要拦截 -->* <**security:http**>  *<!-- 配置剩余的请求地址都需要ROLE\_ADMIN角色才可以访问 -->* <**security:intercept-url pattern="/\*\*" access="hasRole('ADMIN')"**/>  *<!-- 配置开启表单登录 -->* <**security:form-login  login-page="/login.html"  username-parameter="username"  password-parameter="password"  default-target-url="/admin/index.html"**  **always-use-default-target="true"  authentication-failure-url="/login.html"**/>  *<!-- 配置跨站请求伪造不可用-->* <**security:csrf disabled="true"**/>  *<!-- 配置头信息，指定框架页策略(iframe子窗口也可以访问)-->* <**security:headers**>  <**security:frame-options policy="SAMEORIGIN"**/>  </**security:headers**>  </**security:http**>   *<!-- 配置认证管理器 -->* <**security:authentication-manager**>  *<!-- 配置认证提供者 -->* <**security:authentication-provider**>  *<!-- 配置用户认证服务(通过用户名密码方式) -->* <**security:user-service**>  *<!-- 配置用户认证信息：用户名、密码、角色-->* <**security:user name="admin"  password="123456"  authorities="ROLE\_ADMIN"**/>  </**security:user-service**>  </**security:authentication-provider**>  </**security:authentication-manager**> </**beans**> |

配置说明：

如果你在系统中使用了框架页<iframe>标签，需要设置框架页的策略为SAMEORIGIN

same origin: 表示该页面可以在相同域名页面的 frame 中显示。

|  |
| --- |
| <**security:headers**>  <**security:frame-options policy="SAMEORIGIN"**/> </**security:headers**> |

### 2.2.2登录页面

修改pinyougou-manager-web/src/main/webapp/login.html页面

|  |
| --- |
| <**form id="loginForm" action="/login" method="post" class="sui-form"**>  <**div class="input-prepend"**>  <**span class="add-on loginname"**></**span**>  <**input name="username" type="text"  placeholder="邮箱/用户名/手机号" class="span2 input-xfat"**/>  </**div**>  <**div class="input-prepend"**>  <**span class="add-on loginpwd"**></**span**>  <**input name="password" type="password"  placeholder="请输入密码" class="span2 input-xfat"**/>  </**div**>  <**div class="setting"**>  <**div id="slider"**>  <**div id="slider\_bg"**></**div**>  <**span id="label"**>>></**span**>  <**span id="labelTip"**>拖动滑块验证</**span**>  </**div**>  </**div**>  <**div class="logined"**>  <**a class="sui-btn btn-block btn-xlarge btn-danger"  onclick="loginForm.submit();"**>登**&nbsp;&nbsp;**录</**a**>  </**div**> </**form**> |

## 2.3显示登录用户名

### 2.3.1后端代码

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller包下创建LoginController.java

|  |
| --- |
| */\*\* 获取登录用户名控制器 \*/* @RestController **public class** LoginController {  */\*\* 显示登录用户名 \*/* @GetMapping(**"/showLoginName"**)  **public** Map<String, String> showLoginName(){  *// 获取登录用户名* String loginName = SecurityContextHolder.*getContext*()  .getAuthentication().getName();  Map<String,String> data = **new** HashMap<>();  data.put(**"loginName"**, loginName);  **return** data;  } } |

### 2.3.2前端代码

在pinyougou-manager-web/src/main/webapp/js/controller目录下创建

indexController.js：

|  |
| --- |
| *// 定义首页控制器* ***app***.controller(**'indexController'**, **function** ($scope, baseService) {  *// 获取登录用户名* $scope.showLoginName = **function** () {  baseService.sendGet(**"/showLoginName"**)  .then(**function**(response){  *// 获取响应数据* $scope.**loginName** = response.**data**.**loginName**;  });  } }); |

在pinyougou-manager-web/src/main/webapp/admin/index.html引入js

|  |
| --- |
| <**script src="/plugins/angularjs/angular.min.js"**></**script**> <**script src="/js/base.js"**></**script**> <**script src="/js/service/baseService.js"**></**script**> <**script src="/js/controller/indexController.js"**></**script**> |

指令部分

|  |
| --- |
| <**body class="hold-transition skin-green sidebar-mini"  ng-app="pinyougou" ng-controller="indexController"  ng-init="showLoginName()"**> |

将页面上的 测试用户 替换成 {{loginName}}

## 2.4退出登录

修改pinyougou-manager-web/src/main/resources/

applicationContext-security.xml文件，添加如下配置：

|  |
| --- |
| *<!-- 配置哪些请求URL需要拦截 -->* <**security:http**>  *<!-- 配置剩余的请求地址都需要ROLE\_ADMIN角色才可以访问 -->* <**security:intercept-url pattern="/\*\*" access="hasRole('ADMIN')"**/>  *<!-- 配置开启表单登录 -->* <**security:form-login  login-page="/login.html"  username-parameter="username"  password-parameter="password"  default-target-url="/admin/index.html"**  **always-use-default-target="true"  authentication-failure-url="/login.html"**/>  *<!-- 配置跨站请求伪造不可用-->* <**security:csrf disabled="true"**/>  *<!-- 配置头部分，指定框架页策略(iframe子窗口也可以访问)-->* <**security:headers**>  <**security:frame-options policy="SAMEORIGIN"**/>  </**security:headers**>  *<!-- 配置退出登录 -->* <**security:logout logout-url="/logout"**/> </**security:http**> |

加此配置后，会自动的产生退出登录的地址/logout,如果你不想用这个地址 ，你也可以定义生成的退出地址以及跳转的页面，配置如下：

|  |
| --- |
| <**security:logout logout-url="" logout-success-url=""**/> |

logout-url：退出的地址，默认为/logout

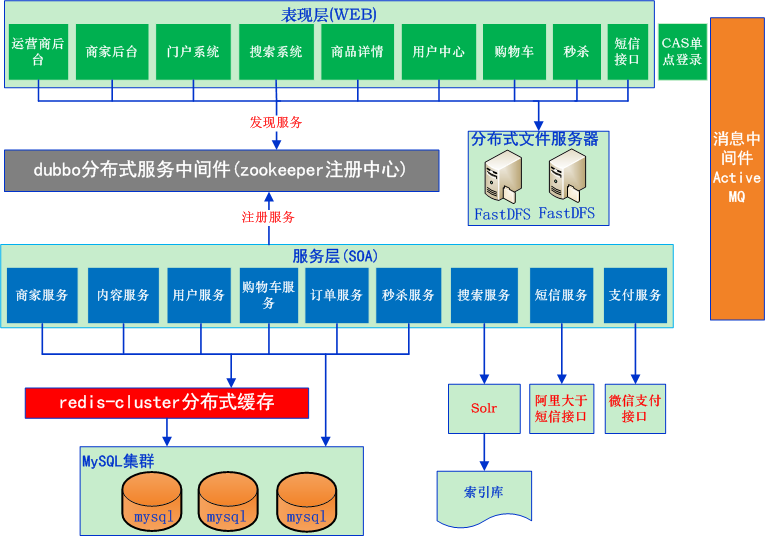
logout-success-url：退出成功后跳转的地址



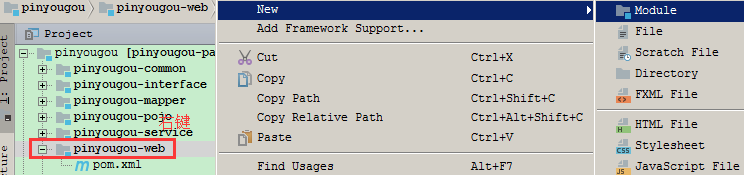
在pinyougou-manager-web/src/main/webapp/admin/index.html页面修改注销的链接：

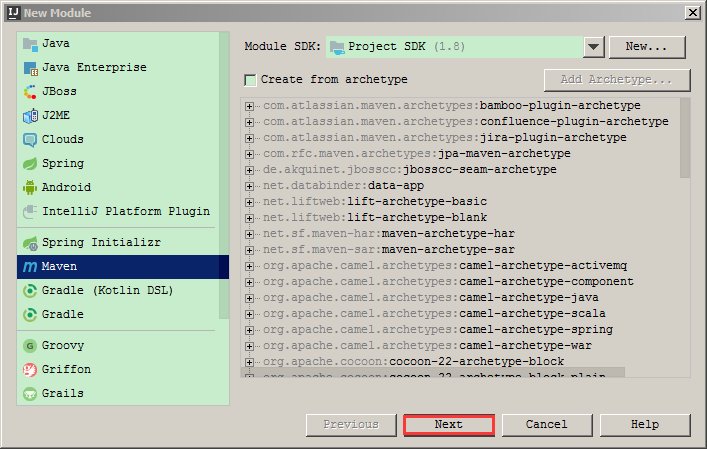
|  |
| --- |
| <**div class="pull-right"**>  <**a href="/logout" class="btn btn-default btn-flat"**>注销</**a**> </**div**> |

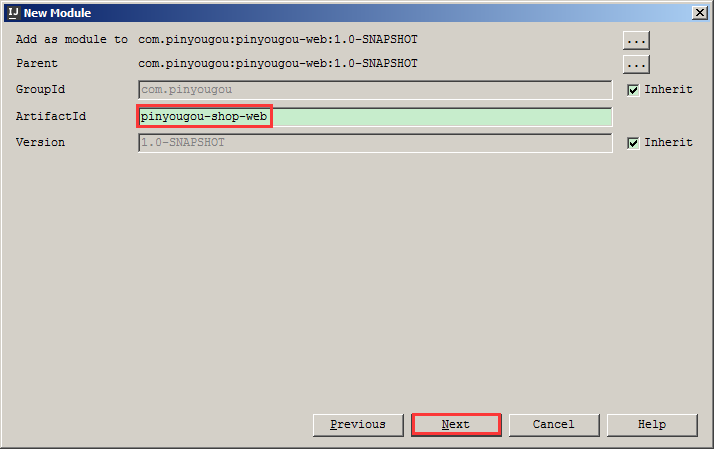
# 3.搭建商家管理后台

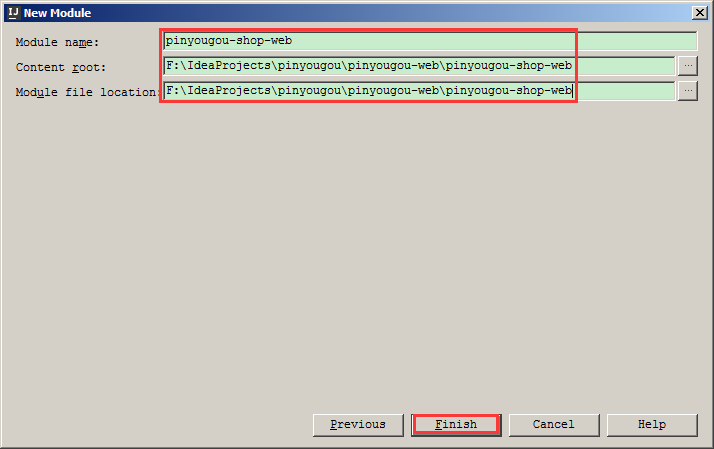


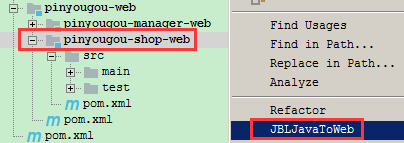
## 3.1 pinyougou-shop-web(war)











与pinyougou-manager-web的构建方式一样，拷贝pinyougou-manager-web模块中的web.xml(去掉SpringSecurity配置)、springmvc.xml、log4j.properties

pinyougou-shop-web/pom.xml：

|  |
| --- |
| <**properties**>  *<!-- 覆盖父工程中的port属性 -->* <**port**>**9102**</**port**> </**properties**> |

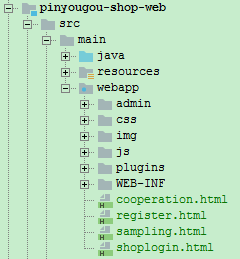
pinyougou-shop-web/src/main/resources/springmvc.xml：

|  |
| --- |
| *<!-- ############## 配置dubbo服务消费者 ############## --> <!-- 配置当前应用的名称 -->* <**dubbo:application name="pinyougou-shop-web"**/> *<!-- 配置zookeeper作为注册中心，注册服务地址 -->* <**dubbo:registry protocol="zookeeper" address="192.168.12.131:2181"**/> *<!-- 配置采用包扫描来引用服务，产生服务接口的代理对象 -->* <**dubbo:annotation package="com.pinyougou.shop.controller"**/> |

创建包名：com.pinyougou.shop.controller

## 3.2准备工作

将“资料\商家管理后台静态资源”下的所有文件拷贝到pinyougou-shop-web/src/main/webapp/目录下



配置域名访问：shop.pinyougou.com

第一步：修改hosts增加

127.0.0.1 shop.pinyougou.com

第二步：修改nginx.conf增加反向代理

|  |
| --- |
| server {  listen 80;  server\_name shop.pinyougou.com;  proxy\_set\_header X-Forwarded-Host $host;  proxy\_set\_header X-Forwarded-Server $host;  location / {  proxy\_pass http://127.0.0.1:9102;  proxy\_connect\_timeout 600;  proxy\_read\_timeout 600;  }  } |

第三步：重启Nginx

nginx -s reload

第四步：访问地址

**http://shop.pinyougou.com/admin/index.html**

# 4.商家申请入驻

## 4.1需求分析

商家申请入驻，需要填写商家相关的信息。待运营商平台审核通过后即可使用。

## 4.3前端代码

修改pinyougou-shop-web/src/main/webapp/register.html页面，引入js

|  |
| --- |
| <**script src="/plugins/angularjs/angular.min.js"**></**script**> <**script src="/js/base.js"**></**script**> <**script src="/js/service/baseService.js"**></**script**> <**script src="/js/controller/baseController.js"**></**script**> <**script src="/js/controller/sellerController.js"**></**script**> |

register.html相关指令：

|  |
| --- |
| <**body ng-app="pinyougou" ng-controller="sellerController"**> |

register.html绑定表单（部分代码）31行-171行：

|  |
| --- |
| <**form class="sui-form form-horizontal"**>  <**div class="control-group"**>  <**label class="control-label"**>登录名（不可修改）：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="登录名"  ng-model="seller.sellerId"  class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>登录密码：</**label**>  <**div class="controls"**>  <**input type="password" placeholder="登录密码"  ng-model="seller.password"  class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>店铺名称：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="店铺名称"  ng-model="seller.nickName"  class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>公司名称：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="公司名称"  ng-model="seller.name" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>公司电话：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="公司电话"  ng-model="seller.telephone" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>公司详细地址：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="公司详细地址"  ng-model="seller.addressDetail" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>联系人姓名：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="联系人姓名"  ng-model="seller.linkmanName" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>联系人QQ：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="联系人QQ"  ng-model="seller.linkmanQq" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>联系人手机：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="联系人手机"  ng-model="seller.linkmanMobile" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>联系人EMAIL：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="联系人EMAIL"  ng-model="seller.linkmanEmail" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>营业执照号：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="营业执照号"  ng-model="seller.licenseNumber" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>税务登记证号：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="税务登记证号"  ng-model="seller.taxNumber" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>组织机构代码证：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="组织机构代码证"  ng-model="seller.orgNumber" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>法定代表人：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="法定代表人"  ng-model="seller.legalPerson" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>法定代表人身份证号：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="法定代表人身份证号"  ng-model="seller.legalPersonCardId" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>开户行名称：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="开户行名称"  ng-model="seller.bankName" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>开户行支行：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="开户行支行"  ng-model="seller.bankUser" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>银行账号：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="银行账号"  ng-model="seller.brief" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>  **&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**</**label**>  <**div class="controls"**>  <**input name="m1" type="checkbox" value="2" checked=""**>  <**span**>同意协议并注册 <**a href="sampling.html"**>《品优购商家入驻协议》</**a**></**span**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**></**label**>  <**div class="controls btn-reg"**>  <**a class="sui-btn btn-block btn-xlarge btn-danger"  ng-click="saveOrUpdate()"**>申请入驻</**a**>  </**div**>  </**div**> </**form**> |

修改pinyougou-shop-web/src/main/webapp/js/controller的

sellerController.js在保存成功后跳转到登录页面

|  |
| --- |
| */\*\* 添加商家 \*/* $scope.saveOrUpdate = **function**(){  */\*\* 发送post请求 \*/* baseService.sendPost(**"/seller/save"**, **$scope.seller**)  .then(**function**(response){  **if** (response.**data**){  */\*\* 跳转到商家登录页面 \*/* ***location***.href = **"/shoplogin.html"**;  }**else**{  *alert*(**"操作失败！"**);  }  }); }; |

## 4.4后端代码

### 4.4.1控制器层

pinyougou-shop-web/src/main/java/com.pinyougou.shop.controller包下创建SellerController.java

|  |
| --- |
| @RestController @RequestMapping(**"/seller"**) **public class** SellerController {  */\*\* 注入商家服务接口代理对象 \*/* @Reference(timeout = 10000)  **private** SellerService **sellerService**;  */\*\* 添加商家 \*/* @PostMapping(**"/save"**)  **public boolean** save(@RequestBody Seller seller){  **try**{  **sellerService**.save(seller);  **return true**;  }**catch** (Exception ex){  ex.printStackTrace();  }  **return false**;  } } |

### 4.4.2服务层

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl包下创建SellerServiceImpl.java

|  |
| --- |
| @Service(interfaceName = **"com.pinyougou.service.SellerService"**) @Transactional **public class** SellerServiceImpl **implements** SellerService {  @Autowired  **private** SellerMapper **sellerMapper**;  */\*\* 添加商家 \*/* @Override  **public void** save(Seller seller) {  **try**{  seller.setStatus(**"0"**);  seller.setCreateTime(**new** Date());  **sellerMapper**.insertSelective(seller);  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  }  }  @Override  **public void** update(Seller seller) {  }  @Override  **public void** delete(Serializable id) {  }  @Override  **public void** deleteAll(Serializable[] ids) {  }  @Override  **public** Seller findOne(Serializable id) {  **return null**;  }  @Override  **public** List<Seller> findAll() {  **return null**;  }  @Override  **public** List<Seller> findByPage(Seller seller, **int** page, **int** rows) {  **return null**;  } } |

**访问地址：http://shop.pinyougou.com/register.html**

# 5.商家审核【运营商】

## 5.1需求分析

商家申请入驻后，需要网站运营人员在运营商后台pinyougou-manager-web中进行审核，审核后商家才可以登录系统。

**状态值： 0：未审核 1：已审核 2：审核未通过 3：关闭**

## 5.2商家待审核列表

### 5.2.1前端代码

修改pinyougou-manager-web/src/main/webapp/admin/seller\_check.html,引入js：

|  |
| --- |
| <**script src="/plugins/angularjs/angular.min.js"**></**script**> *<!-- 分页模块开始 -->* <**script src="/plugins/angularjs/pagination.js"**></**script**> <**link rel="stylesheet" href="/plugins/angularjs/pagination.css"**/> *<!-- 分页模块结束 -->* <**script src="/js/base-pagination.js"**></**script**> <**script src="/js/service/baseService.js"**></**script**> <**script src="/js/controller/baseController.js"**></**script**> <**script src="/js/controller/sellerController.js"**></**script**> |

seller\_check.html相关指令：

|  |
| --- |
| <**body class="hold-transition skin-red sidebar-mini"  ng-app="pinyougou"  ng-controller="sellerController"  ng-init="searchEntity.status=0"**> |

seller\_check.html加入分页指令，</table>标签后面：

|  |
| --- |
| <**tm-pagination conf="paginationConf"**/> |

ng-repeat循环迭代：

|  |
| --- |
| <**tr ng-repeat="entity in dataList"**>  <**td**><**input type="checkbox"**></**td**>  <**td**>{{**entity**.sellerId}}</**td**>  <**td**>{{**entity**.**name**}}</**td**>  <**td**>{{**entity**.nickName}}</**td**>  <**td**>{{**entity**.linkmanName}}</**td**>  <**td**>{{**entity**.telephone}}</**td**>  <**td class="text-center"**>  <**button type="button" class="btn bg-olive btn-xs"  data-toggle="modal" data-target="#sellerModal"**>详情  </**button**>  </**td**> </**tr**> |

seller\_check.htm查询条件、查询按钮绑定点击事件：

|  |
| --- |
| <**div class="has-feedback"**>  公司名称：<**input ng-model="searchEntity.name"**/>  店铺名称： <**input ng-model="searchEntity.nickName"**/>  <**button class="btn btn-default" ng-click="reload()"**>查询</**button**> </**div**> |

### 5.2.2后台代码

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller包下创建SellerController.java：

|  |
| --- |
| @RestController @RequestMapping(**"/seller"**) **public class** SellerController {  @Reference(timeout = 10000)  **private** SellerService **sellerService**;  */\*\* 多条件分页查询商家 \*/* @GetMapping(**"/findByPage"**)  **public** PageResult findByPage(Seller seller, Integer page, Integer rows){  **try** {  */\*\* GET请求中文转码 \*/* **if** (seller != **null** && StringUtils.*isNoneBlank*(seller.getName())) {  seller.setName(**new** String(seller.getName()  .getBytes(**"ISO8859-1"**), **"UTF-8"**));  }  **if** (seller != **null** && StringUtils.*isNoneBlank*(seller.getNickName())) {  seller.setNickName(**new** String(seller.getNickName()  .getBytes(**"ISO8859-1"**), **"UTF-8"**));  }  }**catch** (Exception ex){  ex.printStackTrace();  }  **return sellerService**.findByPage(seller, page, rows);  } } |

pinyougou-interface/src/main/java/com.pinyougou.service/SellerService.java：

|  |
| --- |
| */\*\* 多条件分页查询 \*/* PageResult findByPage(Seller seller, **int** page, **int** rows); |

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl的SellerServiceImpl.java：

|  |
| --- |
| */\*\* 多条件分页查询商家 \*/* @Override **public** PageResult findByPage(Seller seller, **int** page, **int** rows) {  **try**{  */\*\* 开始分页 \*/* PageInfo<Seller> pageInfo = PageHelper.*startPage*(page, rows)  .doSelectPageInfo(**new** ISelect() {  @Override  **public void** doSelect() {  **sellerMapper**.findAll(seller);  }  });  **return new** PageResult(pageInfo.getTotal(), pageInfo.getList());  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  } } |

pinyougou-mapper/src/main/java/com.pinyougou.mapper/SellerMapper.java：

|  |
| --- |
| */\*\* 多条件查询商家 \*/* List<Seller> findAll(Seller seller); |

pinyougou-mapper/src/main/resources/mappers/SellerMapper.xml：

|  |
| --- |
| *<!-- 多条件查询商家 -->* <**select id="findAll" resultType="Seller"**>  select *\** from tb\_seller  <**where**>  <**if test="status != null and status !=''"**>  status = #{status}  </**if**>  <**if test="name != null and name !=''"**>  and name like CONCAT('%',#{name},'%')  </**if**>  <**if test="nickName != null and nickName !=''"**>  and nick\_name like CONCAT('%',#{nickName},'%')  </**if**>  </**where**> </**select**> |

## 5.3商家详情



pinyougou-manager-web/src/main/webapp/admin/seller\_check.html列表的“详情”按钮，绑定点击事件

|  |
| --- |
| <**button type="button" class="btn bg-olive btn-xs"  data-toggle="modal" data-target="#sellerModal"  ng-click="show(entity);"**>详情</**button**> |

pinyougou-manager-web/src/main/webapp/admin/seller\_check.html绑定页面弹出窗口 (107行-198行)

|  |
| --- |
| *<!-- 选项卡开始 -->* <**div id="myTabContent" class="tab-content"**>  <**div class="tab-pane active in" id="home"**>  <**br**/>  <**table class="table table-bordered table-striped" width="800px"**>  <**tr**>  <**td**>公司名称</**td**>  <**td**>{{entity.**name**}}</**td**>  </**tr**>  <**tr**>  <**td**>公司手机</**td**>  <**td**>{{entity.mobile}}</**td**>  </**tr**>  <**tr**>  <**td**>公司电话</**td**>  <**td**>{{entity.telephone}}</**td**>  </**tr**>  <**tr**>  <**td**>公司详细地址</**td**>  <**td**>{{entity.addressDetail}}</**td**>  </**tr**>  </**table**>  </**div**>  <**div class="tab-pane fade" id="linkman"**>  <**br**/>  <**table class="table table-bordered table-striped"**>  <**tr**>  <**td**>联系人姓名</**td**>  <**td**>{{entity.linkmanName}}</**td**>  </**tr**>  <**tr**>  <**td**>联系人QQ</**td**>  <**td**>{{entity.linkmanQq}}</**td**>  </**tr**>  <**tr**>  <**td**>联系人手机</**td**>  <**td**>{{entity.linkmanMobile}}</**td**>  </**tr**>  <**tr**>  <**td**>联系人E-Mail</**td**>  <**td**>{{entity.linkmanEmail}}</**td**>  </**tr**>  </**table**>  </**div**>  <**div class="tab-pane fade" id="certificate"**>  <**br**>  <**table class="table table-bordered table-striped"**>  <**tr**>  <**td**>营业执照号</**td**>  <**td**>{{entity.licenseNumber}}</**td**>  </**tr**>  <**tr**>  <**td**>税务登记证号</**td**>  <**td**>{{entity.taxNumber}}</**td**>  </**tr**>  <**tr**>  <**td**>组织机构代码证号</**td**>  <**td**>{{entity.orgNumber}}</**td**>  </**tr**>  </**table**>  </**div**>  <**div class="tab-pane fade" id="ceo"**>  <**br**>  <**table class="table table-bordered table-striped"**>  <**tr**>  <**td**>法定代表人</**td**>  <**td**>{{entity.legalPerson}}</**td**>  </**tr**>  <**tr**>  <**td**>法定代表人身份证号</**td**>  <**td**>{{entity.legalPersonCardId}}</**td**>  </**tr**>  </**table**>  </**div**>  <**div class="tab-pane fade" id="bank"**>  <**br**>  <**table class="table table-bordered table-striped"**>  <**tr**>  <**td**>开户行名称</**td**>  <**td**>{{entity.bankName}}</**td**>  </**tr**>  <**tr**>  <**td**>开户行支行</**td**>  <**td**>{{entity.bankUser}}</**td**>  </**tr**>  <**tr**>  <**td**>银行账号</**td**>  <**td**>{{entity.brief}}</**td**>  </**tr**>  </**table**>  </**div**> </**div**> *<!-- 选项卡结束 -->* |

## 5.4商家审核

### 5.4.1前端代码

seller\_check.html页面上的按钮，绑定点击事件：

|  |
| --- |
| <**button class="btn btn-success" data-dismiss="modal"  ng-click="updateStatus(entity.sellerId,'1')"  aria-hidden="true"**>审核通过</**button**> <**button class="btn btn-danger" data-dismiss="modal"  ng-click="updateStatus(entity.sellerId,'2')"  aria-hidden="true"**>审核未通过</**button**> <**button class="btn btn-danger" data-dismiss="modal"  ng-click="updateStatus(entity.sellerId,'3')"  aria-hidden="true"**>关闭商家</**button**> <**button class="btn btn-default" data-dismiss="modal"  aria-hidden="true"**>关闭</**button**> |

修改pinyougou-manager-web/src/main/webapp/js/controller的

sellerController.js：

|  |
| --- |
| */\*\* 修改商家状态 \*/* $scope.updateStatus = **function**(sellerId,status){  baseService.sendGet(**"/seller/updateStatus?sellerId="** + sellerId +**'&status='** + status)  .then(**function**(response){  **if**(response.data){  */\*\* 重新加载数据 \*/* $scope.reload();  }**else**{  *alert*(**"操作失败！"**);  }  }); }; |

### 5.4.2后端代码

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller的SellerController.java

|  |
| --- |
| */\*\* 审核商家(修改商家状态) \*/* @GetMapping(**"/updateStatus"**) **public boolean** updateStatus(String sellerId, String status){  **try** {  **sellerService**.updateStatus(sellerId, status);  **return true**;  } **catch** (Exception e) {  e.printStackTrace();  **return false**;  } } |

pinyougou-interface/src/main/java/com.pinyougou.service/SellerService.java服务接口

|  |
| --- |
| */\*\* 修改商家状态 \*/* **void** updateStatus(String sellerId, String status); |

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl的SellerServiceImpl.java服务实现

|  |
| --- |
| */\*\* 修改商家状态 \*/* **public void** updateStatus(String sellerId, String status){  **try**{  Seller seller = **new** Seller();  seller.setSellerId(sellerId);  seller.setStatus(status);  **sellerMapper**.updateByPrimaryKeySelective(seller);  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  } } |

# 6.商家系统登录与安全控制

## 6.1需求分析

完成商家系统登录与安全控制，商家账号来自数据库，并实现密码加密

## 6.2配置与登录页面

1. pom.xml、web.xml、shoplogin.html 参照运营商管理后台

配置spring-security依赖jar包、过滤器、页面表单。

1. pinyougou-shop-web/src/main/resources/applicationContext-security.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:security="http://www.springframework.org/schema/security"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/security  http://www.springframework.org/schema/security/spring-security.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd"**>   *<!--########## 配置请求URL拦截规则 ##########-->  <!-- 配置哪些请求URL不被拦截 -->* <**security:http pattern="/\*.html" security="none"**/>  <**security:http pattern="/css/\*\*" security="none"**/>  <**security:http pattern="/img/\*\*" security="none"**/>  <**security:http pattern="/js/\*\*" security="none"**/>  <**security:http pattern="/plugins/\*\*" security="none"**/>  <**security:http pattern="/seller/save" security="none"**/>   *<!-- 配置哪些请求URL需要拦截 -->* <**security:http**>  *<!-- 配置剩余的请求地址都需要ROLE\_SELLER角色才可以访问 -->* <**security:intercept-url pattern="/\*\*" access="hasRole('SELLER')"**/>  *<!-- 配置开启表单登录 -->* <**security:form-login  login-page="/shoplogin.html"  default-target-url="/admin/index.html"**  **always-use-default-target="true"  authentication-failure-url="/shoplogin.html"**/>  *<!-- 配置跨站请求伪造不可用-->* <**security:csrf disabled="true"**/>  *<!-- 配置头信息，指定框架页策略(iframe子窗口也可以访问)-->* <**security:headers**>  <**security:frame-options policy="SAMEORIGIN"**/>  </**security:headers**>  *<!-- 配置退出登录 -->* <**security:logout logout-url="/logout"**/>  </**security:http**>   *<!-- 配置认证管理器 -->* <**security:authentication-manager**>  *<!-- 配置认证提供者 -->* <**security:authentication-provider  user-service-ref="userDetailsService"**/>  </**security:authentication-manager**>  *<!-- 配置自定义的认证服务 -->* <**bean id="userDetailsService"  class="com.pinyougou.shop.service.UserDetailsServiceImpl"**/> </**beans**> |

## 6.3自定义认证类

在pinyougou-shop-web/src/main/java/com.pinyougou.shop.service包下，创建UserDetailsServiceImpl.java认证类实现UserDetailsService接口

|  |
| --- |
| **package** com.pinyougou.shop.service;  **import** org.springframework.security.core.GrantedAuthority; **import** org.springframework.security.core.authority.SimpleGrantedAuthority; **import** org.springframework.security.core.userdetails.User; **import** org.springframework.security.core.userdetails.UserDetails; **import** org.springframework.security.core.userdetails.UserDetailsService; **import** org.springframework.security.core.userdetails.UsernameNotFoundException; **import** java.util.ArrayList; **import** java.util.List;  */\*\* 用户认证服务类 \*/* **public class** UserDetailsServiceImpl **implements** UserDetailsService {  @Override  **public** UserDetails loadUserByUsername(String username)  **throws** UsernameNotFoundException {  */\*\* 创建List集合封装角色 \*/* List<GrantedAuthority> grantedAuths = **new** ArrayList<>();  */\*\* 添加角色 \*/* grantedAuths.add(**new** SimpleGrantedAuthority(**"ROLE\_SELLER"**));  */\*\* 返回用户信息对象 \*/* **return new** User(username,**"123456"**, grantedAuths);  } } |

说明：经过上述配置，用户在输入密码123456时就会通过（用户名随意）

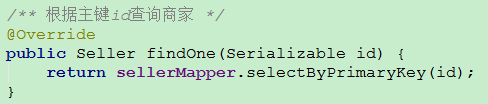
## 6.4认证类调用服务

修改pinyougou-shop-web/src/main/java/com.pinyougou.shop.service的

UserDetailsServiceImpl.java：

|  |
| --- |
| **public** **class** UserDetailsServiceImpl **implements** UserDetailsService {  /\*\* 注入商家服务接口代理对象 \*/  **private** SellerService sellerService;  @Override  **public** UserDetails loadUserByUsername(String username) **throws** UsernameNotFoundException {  /\*\* 创建List集合封装角色 \*/  List<GrantedAuthority> grantedAuths = **new** ArrayList<>();  /\*\* 添加角色 \*/  grantedAuths.add(**new** SimpleGrantedAuthority("ROLE\_SELLER"));  /\*\* 根据登录名查询商家 \*/  Seller seller = sellerService.findOne(username);  /\*\* 判断商家是否为空 并且 商家已审核 \*/  **if** (seller != **null** && seller.getStatus().equals("1")){  /\*\* 返回用户信息对象 \*/  **return** **new** User(username, seller.getPassword(), grantedAuths);  }  return null;  }  **public** **void** setSellerService(SellerService sellerService) {  **this**.sellerService = sellerService;  }  } |

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl的SellerServiceImpl.java



pinyougou-shop-web/src/main/resources/applicationContext-security.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:security="http://www.springframework.org/schema/security"  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/security  http://www.springframework.org/schema/security/spring-security.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd"**>   *<!--########## 配置请求URL拦截规则 ##########-->  <!-- 配置哪些请求URL不被拦截 -->* <**security:http pattern="/\*.html" security="none"**/>  <**security:http pattern="/css/\*\*" security="none"**/>  <**security:http pattern="/img/\*\*" security="none"**/>  <**security:http pattern="/js/\*\*" security="none"**/>  <**security:http pattern="/plugins/\*\*" security="none"**/>  <**security:http pattern="/seller/save" security="none"**/>   *<!-- 配置哪些请求URL需要拦截 -->* <**security:http use-expressions="false"**>  *<!-- 配置剩余的请求地址都需要ROLE\_SELLER角色才可以访问 -->* <**security:intercept-url pattern="/\*\*" access="hasRole('SELLER')"**/>  *<!-- 配置开启表单登录 -->* <**security:form-login login-page="/shoplogin.html"  default-target-url="/admin/index.html"**  **always-use-default-target="true"  authentication-failure-url="/shoplogin.html"**/>  *<!-- 配置跨站请求伪造不可用-->* <**security:csrf disabled="true"**/>  *<!-- 配置头部分，指定框架页策略(iframe子窗口也可以访问)-->* <**security:headers**>  <**security:frame-options policy="SAMEORIGIN"**/>  </**security:headers**>  *<!-- 配置退出登录 -->* <**security:logout logout-url="/logout"**/>  </**security:http**>   *<!-- 配置认证管理器 -->* <**security:authentication-manager**>  *<!-- 配置认证提供者 -->* <**security:authentication-provider  user-service-ref="userDetailsService"**/>  </**security:authentication-manager**>   *<!-- ############## 配置dubbo服务消费者 ############## -->  <!-- 配置当前应用的名称 -->* <**dubbo:application name="pinyougou-shop-web"**/>  *<!-- 配置zookeeper作为注册中心，发现服务地址 -->* <**dubbo:registry address="zookeeper://192.168.12.131:2181"**/>  *<!-- 配置引用单个服务，产生服务接口代理对象 -->* <**dubbo:reference id="sellerService"**  **interface="com.pinyougou.service.SellerService"**/>   *<!-- 配置自定义的认证服务 -->* <**bean id="userDetailsService"  class="com.pinyougou.shop.service.UserDetailsServiceImpl"**>  <**property name="sellerService" ref="sellerService"**/>  </**bean**> </**beans**> |

**说明：经过上述修改后，在登录页面输入用户名和密码与数据库一致即可登录。**

## 6.5密码加密

### 6.5.1 BCrypt加密算法

用户表的密码通常使用MD5等不可逆算法加密后存储，为了防止破解会先使用一个特定的字符串（如域名）加密，然后再使用一个随机的salt（盐值）加密。 特定字符串是程序代码中固定的，salt是每个密码单独随机，一般给用户表加一个字段单独存储，比较麻烦。 BCrypt算法将salt随机并混入最终加密后的密码，验证时也无需单独提供之前的salt，从而无需单独处理salt问题。

### 6.5.2商家入驻密码加密

商家申请入驻的密码要使用BCrypt算法进行加密存储，修改pinyougou-shop-web/src/main/java/com.pinyougou.shop.controller的

SellerController.java的save方法

|  |
| --- |
| */\*\* 添加商家 \*/* @PostMapping(**"/save"**) **public boolean** save(@RequestBody Seller seller){  **try**{  */\*\* 密码加密 \*/* BCryptPasswordEncoder passwordEncoder = **new** BCryptPasswordEncoder();  String password = passwordEncoder.encode(seller.getPassword());  seller.setPassword(password);  **sellerService**.save(seller);  **return true**;  }**catch**(Exception ex){  ex.printStackTrace();  }  **return false**; } |

### 6.5.3加密配置

pinyougou-shop-web/src/main/resources/applicationContext-security.xml

|  |
| --- |
| *<!-- 配置加密对象 -->* <**bean id="bcryptEncoder" class="org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder"**/> |

修改认证管理器

|  |
| --- |
| *<!-- 配置认证管理器 -->* <**security:authentication-manager**>  *<!-- 配置认证提供者 -->* <**security:authentication-provider user-service-ref="userDetailsService"**>  *<!-- 设置加密方式 -->* <**security:password-encoder ref="bcryptEncoder"**/>  </**security:authentication-provider**> </**security:authentication-manager**> |

## 6.6显示登录名

说明：pinyougou-shop-web参照运营商后台(pinyougou-manager-web)

|  |
| --- |
| **LoginController.java：**  @RestController **public class** LoginController {  */\*\* 显示登录用户名 \*/* @GetMapping(**"/showLoginName"**)  **public** Map<String, String> showLoginName(){  *// 获取登录用户名* String loginName = SecurityContextHolder.*getContext*()  .getAuthentication().getName();  Map<String,String> data = **new** HashMap<>();  data.put(**"loginName"**, loginName);  **return** data;  } }  **js/controller/indexController.js：**  *// 定义首页控制器* ***app***.controller(**'indexController'**, **function** ($scope, baseService) {  *// 获取登录用户名* $scope.showLoginName = **function** () {  baseService.sendGet(**"/showLoginName"**)  .then(**function**(response){  *// 获取响应数据* $scope.**loginName** = response.**data**.**loginName**;  });  } });  **admin/index.jsp：**  <**script src="/plugins/angularjs/angular.min.js"**></**script**> <**script src="/js/base.js"**></**script**> <**script src="/js/service/baseService.js"**></**script**> <**script src="/js/controller/indexController.js"**></**script**>  <**body class="hold-transition skin-blue sidebar-mini"  ng-app="pinyougou" ng-controller="indexController"  ng-init="showLoginName()"**>  **把 chendahai 替换成 {{loginName}}** |

## 6.7退出登录

说明：pinyougou-shop-web参照运营商后台(pinyougou-manager-web)

|  |
| --- |
| **admin/index.html页面：**  <**div class="pull-right"**>  <**a href="/logout" class="btn btn-default btn-flat"**>注销</**a**> </**div**> |